In the Claims:

Please amend the claims as follows:

1. (Previously Presented) A method for maximizing CPU performance in a multiprocessor comprising:

configuring a computer system with multiple processors with a common physical memory address space accessible by all processors, and non-shared memory local to a processor;

organizing data elements stored in a shared resource designed to support data manipulation functions, wherein said shared resource is stored in shared memory; and

pipelining execution of an operation with software instructions by partitioning the operating into a series of sequential steps, said instructions comprising:

executing write operations in memory local to a processor in an arbitrary order and at any time prior to storing a pointer from an existing element of said shared resource stored on a computer readable medium to a new element of said shared resource, wherein said pointer is stored in said shared resource;

explicitly indicating a set of write operations to non-local memory be conducted in a specified order; and

executing said write operations to non-local memory prior to storing said pointer from said existing element of said shared resource to said new element of said shared resource in response to said indicating.

2. (Previously Presented) The method of claim 1, further comprising assigning first and second registers of a CPU for storing associated first and second instruction addresses.

3. (Previously Presented) The method of claim 2, further comprising providing a third instruction referencing said registers.

4. (Original) The method of claim 3, wherein said third instruction specifies ordering between said first and second instructions.

5. (Original) The method of claim 4, wherein said third instruction indicates said first instruction's execution attaining a first specified state of execution prior to said second instruction's execution attaining a second specified state of execution.

6. (Original) The method of claim 5, wherein said first and said second specified states of execution are selected from the group consisting of: committing instruction execution, initiating memory access, completing a memory access, initiating an I/O access, completing an I/O access, and completing instruction execution.

7. (Previously Presented) The method of claim 1, further comprising assigning a sequence number to an associated instruction for maintaining instruction ordering.

8. (Previously Presented) The method of claim 7, further comprising statically encoding said sequence number within said instruction.

9. (Previously Presented) The method of claim 7, further comprising dynamically encoding said sequence number within said instruction.

10. (Previously Presented) The method of claim 1, further comprising placing a range of instructions into a hierarchical ordering system.

11. (Previously Presented) The method of claim 10, further comprising implementing a special instruction for maintaining a hierarchical execution of said instruction.

12. Cancel

13. Cancel

14. Cancel

15. Cancel

16. Cancel

17. Cancel

18. Cancel

19. Cancel

20. Cancel

21. Cancel

22. Cancel

23. Cancel

24. Cancel

25. Cancel